



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/163,041	09/29/1998	DANIEL P. VEDITZ	06975-721001	9527
26171 7590 02/06/2008 FISH & RICHARDSON P.C. P.O. BOX 1022 MINNEAPOLIS, MN 55440-1022			EXAMINER FRINK, JOHN MOORE	
			ART UNIT 2142	PAPER NUMBER
			MAIL DATE 02/06/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/163,041

Applicant(s)

VEDITZ, DANIEL P.

Examiner

John M. Frink

Art Unit

2142

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 27 December 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 33-72 and 74-98 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 33-72 and 74-98 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Specification

1. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter for the reasons given below in the 35 USC 112 written description rejection. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o).

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 33, 36, 44, 46, 49, 56, 58, 60, 61, 68, 69, 75 and 82 – 92 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Specifically, claims 33, 36, 44, 46, 49, 56, 58, 60, 61, 68, 69, 75 all reference a 'root archive'. Said 'root' does not appear in any form in the specification, nor in the previous formations of the claims. In order to examine the current claims, 'root' was interpreted to mean the top level of a directory hierarchy.

3. Additionally, claims 33, 46, 60, 68, 75 and 82 – 92, 94 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. Said claims reference a 'nested archive.' No references to said 'nested archive' appear in the specification or in the previous formations of the claims. In order to examine the

current claims, 'nested archive' was interpreted to mean an archive/file/file structure that references another archive/file/file structure.

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 75 – 81 and 91 - 92 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter; said claims are directed to a web browser.
3. A clear disclaimer is requested regarding claim 68, directed toward 'tangible computer readable medium', to verify that Applicant is not intending to claim information received via a network connection.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 33 – 72 and 74 - 98 are rejected under 35 U.S.C. 102(b) as being anticipated and by the Applicant's product 'Netscape Smartupdate' (product information and release notes in 'Netscape Launches Navigator 4.0 With Netcaster, Expanding Line of Client Products', hereafter Press Release 1, and 'Netscape and Lotus Partner to

Offer eSuite Applets Through Netscape Netcenter Online Service', hereafter Press Release 2, with further elaboration of Netscape Smartupdate in 'locally install java applet support?' news group posting by A. Spyker, hereafter Spyker, specifically detailing how Applicant's product supports the nesting of Java JAR files).

Applicant's 'Netscape Smartupdate' product was released and distributed publicly more than one year before Applicant's filing date (as shown by Press Release 1).

3. Claims 33 – 34, 36 – 39, 41 – 42, 44, 45 – 47, 49 – 52, 54 - 55, 57 - 64, 66 – 70, 72, 74, 83, 86, 88, 90 and are rejected under 35 U.S.C. 102(b) as anticipated by Englander (Developing Java Beans).

4. Regarding claims 33 and 46, Englander shows a method for executing an application that is encapsulated in a package, and, Englander shows a computer program product for executing an application that is encapsulated in a package, the computer program product being embodied in a computer readable medium and including instructions that, when executed by a processor, cause the processor to:

load the package (represented by the HTML file with the containing the <APPLET>) within a browser on a local client computer (Section 1.2.2), the package including a manifest (represented by the contents of the <APPLET> tag, Section 6.3.1) and a root archive of files (represented by the content after 'ARCHIVE =' of 6.3.1) that include instructions and content needed to execute the application, the root archive of files including:

an initial file that includes instructions for initiating execution of the application (represented by PickleUser.class) and

other files needed to execute the application (represented by BeansBook.jar)

a nested archive of additional files, the nested archive being included within the package (represented by BeansBook.jar) and

the manifest including an initial file identifier that indicates that the initial file is to be processed before the other files in the root archive of files when the application is executed (represented by the <APPLET> tag, specifically the CODE section that points to PickleUser.class; Section 6.3.1);

in response to loading the package within the browser, automatically accessing the manifest; locating the initial file identifier in the manifest; based on the located initial file identifier, accessing the instructions for initiating the execution of the application from the initial file; processing the accessed instructions from the initial file (where PickleUser.class is a compiled Java file contains instructions in the form of machine-readable execution code); and automatically initiating execution of the application based on the processed instructions (represented by automatically loading and executing PickleUser.class; Section 6.3.1);.

5. Regarding claims 34 and 47, Englander further shows the method for executing an application that is encapsulated in a package of claim 33, further comprising receiving the package at the local client computer (6.3.1, where said HTML file and the code and files within the <APPLET> Tags are received at the client).

6. Regarding claims 36 and 49, Englander further shows accessing the additional instructions in the one or more other files (6.3.1, for example, SourPickle.ser); and

processing the additional instructions, the accessing and processing of the additional instructions being performed in response to processing the instructions for initiating execution of the application from the initial file (where PickleUser.class, the initial file, utilizes/references other files within BeansBook.jar).

7. Regarding claims 37 and 50, Englander further shows where the initial file comprises a source file for a web page, the instructions for initiating execution of the application from the initial file comprise instructions for rendering the web page, and automatically initiating execution of the application comprises rendering the web page in accordance with the instructions for rendering the web page (6.3 and 6.3.1, where the HTML files shown comprise a standard web page source file, which is inherently rendered when it is run).

8. Regarding claims 38 and 51, Englander further shows where the source file for the web page comprises an HTML document (6.3 and 6.3.1).

9. Regarding claims 39 and 52, Englander further shows where the initial file comprises an executable file, and the instructions for initiating execution of the application from the initial file comprise program execution instructions (6.3.1, where PickleUser.class, contained in BeansBook.jar, is referenced. Furthermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

10. Regarding claims 41 and 54, Englander further shows where the instructions for initiating execution of the application from the initial file comprise JAVA-based instructions (6.3.1, where BeansBook.jar contains the Java-based instructions in PickleUser.class).

11. Regarding claims 42 and 55, Englander further shows where the manifest further comprises an archive type identifier that identifies an application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file of identified as being of the type '.class'), and processing the accessed instructions comprises processing the accessed instructions in accordance with the application type of the application (6.3.1, where appletviewer runs files in accordance with types .html, and .jar).

12. Regarding claims 44 and 58, Englander further shows where the initial file identifier indicates that the initial file is the first file to be processed in the archive of files when the application is executed (6.3.1, paragraphs 3 – 4).

13. Regarding claim 57, Englander further shows where the computer program product comprises a browser (Section 1.2.2).

14. Regarding claims 45 and 59, Englander further shows where automatically initiating execution of the application based on the processed instructions comprises automatically initiating execution of the application without maintaining a connection between the local client computer and a web server (6.3.1 where the .jars are downloaded, 6.4 where downloaded .jars are executed without requiring said connection as all required items are on users local "C:" drive).

15. Regarding claim 60, Englander shows method for encapsulating an application in a package such that the application may be automatically executed by a browser (Section 1.2.2) on a client computer, and, a self-contained package for distributing an application, the package being embodied in a computer readable medium (represented as the directory holding the referenced HTML file and the referenced .jar file; Section 6.3.1), and configured to enable the application to be automatically executed within a browser environment (1.2.2), the package comprising:

- generating a root archive of files (represented by the data after 'ARCHIVE =') that include instructions and content needed to execute the application, the archive including: an initial file that includes instructions for initiating execution of the application (represented by PickleUser.class), and other files needed to execute the application (represented by BeansBook.jar) and

- a nested archive that includes additional files (represented BeansBook.jar)
 - generating a manifest file that is associated with the archive, the manifest file including an initial file identifier that instructs the browser to process the initial file before processing other files in the root archive in order to initiate execution of the application (represented by the CODE element of the <APPLET> tag, which specifies executing the PickleUser.class file before any others in order to initiate execution of the BeansBook.jar application) and

- encapsulating the root archive of files, including the nested archive, and the manifest file within a package (where the package is the HTML file holding the <APPLET> And the data within it)

16. Regarding claim 61, Englander further shows where the instructions for initiating execution of the application from the initial file include instructions to process additional instructions from one or more of the other files in the root archive (6.1, 6.2 and 6.3.1, where multiple .jar files can be used, and where each .jar has, by default, a Manifest file specifying additional instructions).

16. Regarding claim 62, Englander further shows where the initial file comprises a source file for a web page, and the instructions for initiating execution of the application from the initial file comprise instructions for rendering the web page (6.3 and 6.3.1, where the HTML files shown comprise a standard web page source file, which is inherently rendered when it is run).

17. Regarding claim 63, Englander further shows where the source file for the web page comprises an HTML document (6.3 and 6.3.1).

18. Regarding claim 64, Englander further shows where the initial file comprises an executable file, and the instructions for initiating execution of the application from the initial file comprise program execution instructions (6.3.1, where PickleUser.class, contained in BeansBook.jar, is referenced. Furthermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

19. Regarding claim 66, Englander further shows where the instructions for initiating execution of the application from the initial file comprise JAVA-based instructions (6.3.1, where BeansBook.jar contains the Java-based instructions in PickleUser.class).

20. Regarding claim 67, Englander further shows where the manifest file further comprises an archive type identifier that instructs the browser to process the

instructions for initiating execution of the application from the initial file in accordance with the application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file identified as being of the type '.class'; and 6.3.1, where appletviewer runs files in accordance with types .html, and .jar).

21. Regarding claim 68, Englander further shows a self-contained package for distributing an application, the package being embodied in a tangible computer readable medium and configured to enable the application to be automatically executed within a browser environment, the package comprising a root archive including (represented by the data after "ARCHIVE =")

- an initial source having instructions for initiating execution of the application within the browser environment (represented by the instructions within PickleUser.class)

- additional files that include instructions and content needed to execute the application within the browser environment (represented by additional items in BeansBook.jar) and

- a nested archive that includes other files, the nested archive being included within the self-contained package (represented by BeansBook.jar)

- a manifest associated with the root archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the root archive when the application is executed (represented by the data within the <APPLET> tags, which indicates that PickleUser.class is the initial content).

22. Regarding claim 69, Englander further shows where said initial content source is an HTML file containing content layout instructions for rendering a document that displays content included in the root archive (6.3 and 6.3.1, where said layout instructions are represented by the WIDTH and HEIGHT elements of the <APPLET> tag).

23. Regarding claim 70, Englander further shows where said initial content source is an executable file containing program execution instructions (6.3.1, where the executable instructions are contained within the <APPLET> tag).

24. Regarding claim 72, Englander further shows where the manifest further comprises an archive type identifier that identifies an application type of the application (6.3.1, where identifies the ARCHIVE as of type '.jar', which further contains a file of identified as being of the type '.class').

25. Regarding claim 74, Englander further shows where the additional files comprise one or more of a web page, a script, an image, a sound file, and a JAVA file (6.2.1).

26. Regarding claims 83, 86, 88 and 90, Englander further shows where the nested archive is configured according to a .JAR type file structure (showing BeansBook.jar nested inside of the HTML file, further containing the <APPLET> tag and "ARCHIVE="). Regarding claim 84, Englander further shows where the nested archive of additional files is one of the other files needed to execute the application (represented by the files required that are in BeansBook.jar, such as SourPickle.ser, among others).

27. Regarding claim 94, Englander further shows establishing security credentials for the initial file, the other files needed to execute the application and the nested archive

using a single signature for the package (6.5).

Claim Rejections - 35 USC § 103

28. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

29. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

30. Claims 35 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Rowe et al. (5,964,836).

Englander shows the method of claim 34 and the computer program product of claim 47.

Englander does not show where receiving the package at a local client computer comprises loading the package onto the local client computer from a local computer readable medium.

Rowe et al. shows where receiving the package at a local client computer comprises loading the package onto the local client computer from a local computer readable medium (col. 6 lines 47 – 50).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Rowe et al. in order to utilize a standard and convenient method of loading an application on a computer.

31. Claims 40, 53, 65 and 71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Bates et al. (5,877,766).

Englander shows the methods of claim 39 and 65, the computer program product of claim 52, and the self-contained package of claim 70.

Englander does not show where the executable file is a JavaScript file.

Bates et al. shows where an executable file is a JavaScript file (col. 6 lines 37 – 59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Bates et al. in order to utilize a standard and common executable file format.

32. Claims 43 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Oran et al. (5,617,526).

Englander shows the methods of claim 33 and the computer program product of claim 45.

Englander does not show displaying an icon associated with the application, wherein loading the package in the browser or other computer program product comprises automatically loading the package in the browser or computer program product in response to a user selecting the icon associated with the application.

Oran et al. shows display an icon associated with an application, wherein loading the package in the browser or other computer program product comprises automatically loading the package in the browser or computer program product in response to a user selecting the icon associated with the application (col. 4 lines 19 – 38).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Oran et al. in order to utilize a standard method of identifying and launching a browser or other computer program product.

29. Claims 75 – 77 and 79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw (6,065,024).

30. Regarding claim 75, Englander shows an application executing an application encapsulated in a self-contained package (represented by the HTML file containing the <APPLET>) comprising: an input module for inputting the self-contained package, wherein the self-contained package includes: a root archive (represented by the text/data after "ARCHIVE =") including

an initial content source having instructions for initiating execution of the application (represented by PickleUser.class), and

additional files that include instructions and content needed to execute the application (represented by BeansBook.jar), and

a manifest associated with the archive that includes an initial content identifier that indicates that the initial content source is to be processed before the additional files in the archive when the application is executed (represented by the data between the <APPLET> tag)

and a nested archive that includes other files, the nested archive being included within the self-contained package (represented by the files in 'Another.jar', which contains additional files, which is rooted in 'ARCHIVE')

a processing engine for accessing the manifest, locating the initial file identifier in the manifest (6.1 – 6.3.1, specifically where the Java runtime environment executes the jar file, or, alternatively, when embedded within HTML, the Java runtime environment is called through the use of 'appletviewer'),

based on the located initial content identifier, accessing the instructions for initiating execution of the application from the initial content source, and processing the instructions for initiating execution of the application from the initial content source before processing the instructions from the additional files (6.1 – 6.3).

Englander does not show where said application for executing said encapsulated application is a web browser, or a content rendering and layout module for rendering content according to the instructions from the initial content source and the additional files.

Renshaw shows where said application for executing said encapsulated application is a web browser, or a content rendering and layout module for rendering content according to the instructions from the initial content source and the additional files (col. 2 lines 9 – 62).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Renshaw in order to utilize a web browsers, applications common at the time of the invention, for executing said Java .jar packages.

31. Regarding claim 76, Englander in view of Renshaw further show where said initial content source is an HTML file containing content layout instructions for rendering

a document that displays content included in the archive (Englander, 6.3.1, where said instructions are the WIDTH and HEIGHT elements within the <APPLET> tag).

32. Regarding claim 77, Englander in view of Renshaw further show where said initial content source is an executable file containing program execution instructions (6.3.1, where PickleUser.class, contained in BeansBook.jar, is referenced. Furthermore, PickleUser.class contains program instructions executable by web browsers and appletviewer).

33. Regarding claim 79, Englander in view of Renshaw further show said executable file includes JAVA-based instructions (6.3.1).

34. Claim 78 is rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw as applied to claim 77 above, and further in view of Bates et al.

Englander in view of Renshaw show the web browser according to claim 77.

Englander in view of Renshaw do not show where said executable file includes Javascript instructions.

Bates et al. show a web browser executing Javascript (col. 6 lines 37 – 59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to further modify the disclosure of Englander with that of Bates et al. in order to support another computer programming language, commonly utilized in web browsers at the time of the invention.

35. Claims 80 and 81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw as applied to claim 75 above, and further in view of Rowe et al.

36. Regarding claim 80, Englander in view of Renshaw show the web browser according to claim 75.

Englander in view of Renshaw do not show where the input module is configured to input the self-contained software package from a computer readable medium.

Rowe et al. shows where the input module is configured to input the self-contained software package from a computer readable medium (col. 6 lines 47 – 50).

It would have been obvious to one of ordinary skill in the art at the time of the invention to further modify the disclosure of Englander with that of Rowe et al. in order to utilize a method that was both standard and convenient at the time of the invention.

37. Regarding claim 81, Englander in view of Renshaw, further in view of Rowe et al. show where said computer readable medium is a compact disc containing digitally recorded information (Rowe et al., col. 6 lines 47 – 50).

38. Claim 93 is rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Neal.

Englander shows claim 34.

Englander does not show receiving an email that includes the package at a local client computer.

Neal shows receiving an email that includes the package at a local client computer (col. 2 lines 18 – 25).

It would have been obvious to one of ordinary skill in the art at the time of the invention to further modify the disclosure of Englander with that of Neal in order to enable additional options for more flexible and adaptable software installation.

39. Claims 82, 85, 87, 89 and 96 and 98 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Applicant's product 'Netscape Smartupdate' (product information and release notes in 'Netscape Launches Navigator 4.0 With Netcaster, Expanding Line of Client Products', hereafter Press Release 1, and 'Netscape and Lotus Partner to Offer eSuite Applets Through Netscape Netcenter Online Service', hereafter Press Release 2, with further elaboration of Netscape Smartupdate in 'locally install java applet support?' news group posting by A. Spyker, hereafter Spyker).

40. Regarding claim 82, Englander shows claim 33. Englander also shows that JAR files contain the claimed initial file identifiers in the JAR's manifest file (6.1, 6.2), that a JAR is a package (6.1), and that JAR contain additional files needed for execution (supporting classes/compiled Java code, supporting graphics, etc., 6.2.1) and that there can be multiple JARs, each containing said initial file identifiers, additional needed files, and each being a package (6.1, 6.3.1).

Englander does not show wherein the nested archive is included within another package, the other package being included within the package and including another initial file that includes instructions for executing the application and another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive when the application is

executed and the method further comprising accessing the other manifest after locating the initial file identifier in the manifest, locating the other initial file identifier in the other manifest and accessing the instructions for executing the application from the other initial file.

Netscape Smartupdate shows wherein the nested archive is included within another package, the other package being included within the package and including another initial file that includes instructions for executing the application and another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive when the application is executed and the method further comprising accessing the other manifest after locating the initial file identifier in the manifest, locating the other initial file identifier in the other manifest and accessing the instructions for executing the application from the other initial file. This claim language is shown through Netscape Smartupdate's support of JAR files inside of other JAR files (Spyker).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the disclosure of Englander with that of Netscape Smartupdate in order to allow for additional flexibility options for packaging applications. Furthermore, the supporting disclosure of Spyker shows that one of ordinary skill in the art was aware of and appreciated the value of being able to put a JAR within a JAR, as was supported by Netscape's Smartupdate product (Spyker, paragraph 4).

41. Regarding claim 85, Englander in view of Netscape Smartupdate further show wherein the nested archive is included within another package, the other package being

included within the package and including another initial file that includes instructions for executing the application and another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive when the application is executed; and the computer program product for executing an application that is included in a package further comprising instructions that when executed by the processor, cause the processes to access the other manifest after locating the initial file identifier in the manifest, locate the other initial file identifier in the other manifest and access the instructions for executing the application from the other initial file (Englander, 6.1, 6.2.1 and 6.3.1).

42. Regarding claim 87, Englander in view of Netscape Smartupdate further show wherein the nested archive is included within another package, the other package being included within the package (as shown by Netscape's Smartupdate product) and including another initial file that includes instructions for executing the application and another manifest that includes another initial file identifier that indicates that the other initial file is to be processed before the additional files in the nested archive when the application is executed (Englander, 6.1 and 6.2.1).

43. Regarding claim 89, Englander in view of Netscape Smartupdate further show wherein the nested archive is included within another self-contained package, the other self-contained package being included within the self-contained package (as shown by Netscape's Smartupdate product, specifically with storing a .JAR inside another .JAR) and including another initial file that includes instructions for executing the application and another manifest that includes another initial file identifier that indicates that the

other initial file is to be processed before the additional files in the nested archive when the application is executed (Englander, 6.1 and 6.2.1).

44. Regarding claim 96, Englander in view of Netscape Smartupdate further show a method for executing an application, the method comprising loading, within a processing environment on a local client computer, a first package that includes a first manifest and a first archive of files (Englander, 6.1 and 6.3.1, where a JAR file is a package contains a manifest and a first archive of files),

the first manifest including a first file identifier that identifies a particular file from within the first archive of files as a first initial file that is to be processed before other files in the first archive of files with the application is executed (Englander, 6.2) and

the first archive of files includes the first initial file, the first initial file including instructions for executing the application (Englander, 6.2, where said first initial file is a compiled Java file (.class file) which contains said instructions)

other files needed to execute the application (Englander, 6.2.1)

a second package that is included within the first package (Netscape Smartupdate) the second package including a second manifest that includes a second initial file identifier that identifies a particular file from within the second archive of files as a second initial file that is to be processed before other files in the second archive of files when the application is executed and a second archive of files that includes the second initial file, the second initial file including instructions for executing the application and additional files needed to execute the application (Englander, 6.2 - 6.3.1)

in response to loading the first package within the processing environment, automatically accessing the first manifest, locating the first initial file identifier in the first manifest, based on the located first initial file identifier, accessing the instructions for initiating execution of the application from the first initial file, processing the accessing instructions from the first initial file and automatically initiating execution of the application based on the processed instructions from the first initial file (Englander, 6.2 and 6.2.1 and 6.3.1).

45. Regarding claim 98, Englander in view of Netscape Smartupdate further show accessing the second manifest after locating the first initial file identifier in the first, locating the second initial file identifier in the second manifest, based on the located second initial file identifier, accessing the instructions for executing the application from the second initial file, processing the accessed instructions from the second initial file and executing the application based on instructions processed from the second initial file (Netscape Smartupdate showing the nested JAR files, Englander 6.1 – 6.3.1 showing JARs including initial file identifiers in the JARs manifest files).

46. Claims 95 and 97 are rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Netscape Smartupdate as applied to claims 82 and 96 above, and further in view of Waldin, Jr. et al. (6,052,531), hereafter Waldin.

Englander in view of Netscape Smartupdate show claims 82 and 96, but do not explicitly state the files are 'executable JAR files.'

Waldin discloses executable JAR files (col. 41 lines 18 – 53).

It would have been obvious to one of ordinary skill in the art at the time of the

invention to modify the disclosure of Englander in view of Netscape Smartupdate with that of Waldin, as Waldin also utilizes JAR files and Waldin is also concerned with software updating (similar to Netscapes Smartupdate). Furthermore, Waldin provides additional guidance and expiation of how to utilize JAR files, increasing the resulting inventions utility.

47. Claims 91 and 92 rejected under 35 U.S.C. 103(a) as being unpatentable over Englander in view of Renshaw as applied to claim 75 above, and further in view of Netscape Smartupdate.

Englander in view of Renshaw show claim 75, including using initial files to execute applications (represented by the .class file specified in the manifest contained within the JAR, Englander, 6.1 - 6.2.1).

Englander in view of Renshaw do not show where the nested archive is included within another self-contained package, the other self-contained package being included with the self-contained package and including another initial file that includes instructions for executing the application and another manifest file that includes another initial file identifier that indicates the other initial file is to be processed before the additional files in the nested archive when the application is executed.

Netscape's Smartupdate shows where the nested archive is included within another self-contained package, the other self-contained package being included with the self-contained package and including another initial file that includes instructions for executing the application and another manifest file that includes another initial file identifier that indicates the other initial file is to be processed before the additional files

in the nested archive when the application is executed, through the disclosure of storing one JAR within another.

Response to Arguments

48. Applicant's arguments filed 12/31/2007 have been fully considered but they are not persuasive.

49. Applicant beings by arguing the 35 USC 112 written description rejections made regarding the use of the terms 'root' and 'nested archive' in the amended claims.

Applicant sites pg. 13, lines 11 - 21 as support for these terms. However, Applicant's cited section does not contain the terms 'root' or 'nested archive', and thus Applicant's arguments are nor persuasive.

50. Applicant next argues the rejections made in view of Englander under 35 USC 102. Applicant's arguments are unpersuasive for the reasons given in the above rejections, which contain a clarified interpretation of the Englander reference in light of the amended claims.

51. Applicant's further arguments rely on Englander not providing proper support for a rejection under 35 USC 102. However, this argument is moot in view of the clarified interpretation of the Englander reference in light of the amended claims, detailed in rejections made in the preceding section.

Conclusion

52. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Nesting CAB Files. microsoft.public.java.cab.

http://groups.google.com/group/microsoft.public.java.cab/browse_thread/thread/5c7c5a123e14dec8/4eb1530cd9d33b01?lnk=st&q=create+cab+within+cab#.

March 6, 1998 - March 25, 1998. pgs. 1 - 7.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John M. Frink whose telephone number is (571) 272-9686. The examiner can normally be reached on M-F 7:30AM - 5:00PM EST; off alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571)272-3868. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Application/Control Number:
09/163,041
Art Unit: 2142

Page 26

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

John Frink
(571) 272-9686


ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER